Music Analytics and Curation: Project Report

2189094T

December 2018

Github URL: https://2189094t.github.io/MCA/index.html

Word Count: 2,589

Contents

Introduction

The aim of this project was to build a dataset that analysed and featured three forms of music-related data: descriptive, notated and acoustic. In building my dataset, I used a variety of techniques and methods to create, analyse and present several forms of data and created many useful resources and interesting insights into different pieces of music. The web-based hosting service Github was then later used to present this data in an accessible and interesting way. In this report, I will detail my methods, evaluate the different data produced, report on any insights gained and identify any improvements I could have made to my dataset with reference to relevant literature.

Dataset Theme

For the cohesive theme of my dataset, I chose the musical works of Adele. I have a particular interest in this artist and she has a wide variety of published material thus I thought it would be very insightful and interesting to analyse the different forms of musical data that currently exist, create new data and produce a large collection of analysis. I also choose this artist as I felt that it would be easy to acquire several current manifestations of all three datatypes.

Insights Gained from Curation Techniques and Analysis

Descriptive Data

Current Manifestations

Descriptive data in this context is metadata attached to musical content which can be used to describe, identify and help catalogue pieces. There are several manifestations of descriptive data that currently exist digitally.

Music streaming platforms (Spotify, iTunes Music and Deezer etc.) store and present a plethora of descriptive data on musical pieces such as track titles, artists, genres, writers and producers which is all typically organised in a very structured way. Many of these services use this data to fundamentally organise their platforms and help the music become more accessible to users. However, some of these streaming platforms also use descriptive data to help their algorithms predict what users like and want to listen to and create playlists of similar music.

Spotify goes slightly further than storing just these basic fields however through their “audio features” criteria. This extends the “genre” information by storing information such as acousticness (a confidence rating on whether a song is acoustic or not), danceability (the extent to which a song is danceable) and instrumentalness (whether a track contains vocals or not)[1]. This data gives the user the power to

[1] “Get Audio Track Features for a Track,” Spotify for Developers, accessed November 28, 2018, https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/.

search using highly specific criteria concerning the actual content of the music rather than a genre that has been attributed to it.

Metadata and MEI Header

In this project, I encoded each MEI document (defined in the Notated Data section below) with metadata in the MEI header to help identify each file. Using several different tags in the header, source, encoding and revision information[2] were encoded in order to enrich the files with context and meaning. Within the webpages storing the scores on Github, code was inserted which displays some of this metadata with links to an authURI (a link to an authority to accredit the information).

Notated Data

MuseScore and Notated Sheet Music

Notated data regarding music typically refers to information representing pitch, rhythm and time but can also include information on lyrics, volume, instruments or performing methods. Within this dataset, a focus was placed on common Western music notation.

In week two, I sourced pieces of notated data for four Adele songs in order to transcribe them into the notation software MuseScore. On the internet, it is very easy to source images or files containing notated data from webpages such as SheetsDaily.com and Musicnotes.com. The four pieces that I sourced were from SheetsDaily.com as the website provides high quality, free sheet music. All four pieces were all composed for piano with two also having lyric information.

MuseScore is a free score writer tool that allows the user to create sheet music easily and export it in several different formats that can later be used to facilitate analysis and presentation. In the second part of the encoding task, I manually transcribed the printed scores into the software in order to turn them into digital files. Whilst this was a time-consuming task, I was able to produce very accurate and useful notated files. I encoded all four pieces in order to amass a larger dataset that could later be analysed.

The digitising of these scores is very useful as it allows them to become computer readable however in using this software, it does not make the score extensible (due to files being software dependant) thus conversion to different forms of encoding is required.

MIDI and MusicXML

In week three of this project, a focus was put on investigating different encoding methodologies which would facilitate the conversion of the notated files into more

[2] “The MEI Header,” MEI, accessed 28 November, 2018, https://music-encoding.org/guidelines/v3/content/header.html.

extensible formats. The first two encoding methods investigated were MIDI and MusicXML.

MIDI (Musical Instrument Digital Interface) is a technical standard that defines a communication protocol, digital interface, and electrical connectors that facilitate the seamless connection of different audio devices[3]. A MIDI file stores a variety of instructions such as position, length, pitch and velocity which can be sent to and understood by MIDI devices to allow them to communicate (i.e. an electronic keyboard and a computer). MusicXML is an open encoding standard that facilitates the digital exchange of sheet music in order to make music more accessible and help archive music for the future[4]. It stores data in XML format with different tags representing pitch, rhythm and time. For the Adele songs, I created MIDI files and MusicXML through exporting them from MuseScore.

In converting my notated files, I found strengths and limitations in both file formats. In MusicXML, whilst this format provides an effective, interoperable representation of musical scores, it omits formatting concepts such as pages and some performance information[5]. It is designed for exchanging representation thus information concerning on/off velocities and dynamics are not considered[6]. With MIDI, whilst it holds information on position, length, pitch and velocity (which allows it to play the same sounds on different systems), it has no concept of individual notes and how they will be represented visually on a score thus making it less useful than MusicXML in representing notated sheet music data.

Music Encoding Initiative (MEI)

MEI (Music Encoding Initiative) is a "community-driven effort to define a system for encoding musical documents in a machine-readable structure.[7]" The MEI govern the interactions of many multidisciplinary parties in order to discuss the best practice for the encoding of music. As with MusicXML, MEI mark-up files are encoded as XML files.

Verovio is an online tool that allows the conversion of MEI or MusicXML music scores into vector graphics and vice versa. For my dataset, I made use of this software to create MEI files that I could use to evaluate MEI encoding in comparison to MusicXML and also to create the visual sheet music for my dataset on Github.

In comparison to MusicXML, it is clear to see that there are many differences to MEI. Firstly, on initial visual inspection, there is a distinct difference in simplicity. MusicXML is much more generalised and visually more user friendly as it contains each note in its own tag with clear attribute terms such as <pitch>, <duration> and <stem>. Ties are also represented with a start tag in the first note and an end tag in

[3] "MIDI," Wikipedia, accessed November 28, 2018, https://en.wikipedia.org/wiki/MIDI.

[4] "MusicXML: Version 3.1," W3C, accessed November 28, 2018, https://www.w3.org/2017/12/musicxml31/.

[5] Michael Good, "MusicXML for Notation and Analysis," Songs and Schemas, accessed November 28, 2018, http://michaelgood.info/publications/music/musicxml-for-notation-and-analysis/.

[6] "Advantage of MIDI," steinberg.help, accessed November 28, 2018, https://steinberg.help/cubase_pro_artist/v9/en/cubase_nuendo/topics/midi_editor_score_editor/working_with_music_xml/score_editor_midi_advantages_of_c.html.

[7] "Music Encoding Initiative," MEI, accessed November 28, 2018, https://music-encoding.org/.

the last note making it easy to identify where the ties are placed. MEI files however contain note information like ties at the end of each bar and not with the individual note information which can be difficult to identify and understand. As MEI is much more precise, it is much harder for developers to work with and read. Also, MEI encoding uses less obvious terms such as “pname” to represent a note. The code below shows the differences in complexity in the way both formats represent the same, single note:

MusicXML	MEI
<note default-x="106.46" default-y="-65.00"> <pitch> <step>G</step> <octave>3</octave> </pitch> <duration>8</duration> <voice>1</voice> <type>half</type> <stem>up</stem> <staff>1</staff> </note>	<note xml:id="note-0000001241270394" oct="3" pname="g" />

In the latter part of Week 3’s task, I edited a couple of my MEI files. The files I edited reflected changes to the songs that Adele frequently makes whilst performing them live (i.e. the first two bars of Hello get doubled). This was a time-consuming task as I had to edit each xml:id to make each element unique and had to change the measure number for each bar individually to make it semantically accurate. Whilst time-consuming, I feel I created very useful versions from the originals.

jSymbolic and music21

Week 5 involved looking at some analytical techniques for encoded notation. For this, the software jSymbolic was used. This software was produced to aid in conducting research in music information retrieval, musicology and music theory and can extract 246 unique features (ranging from Mean Pitch to Range to Most Common Melodic Interval) from either MIDI or MEI files[8]. Using this software, I created several different CSV files containing interesting, relevant analysis for the four Adele songs. In my dataset on GitHub, I made the raw data CSV files available but also created an Excel document where I created graphs and put the extracted feature data into context.

Further analysis on the data was also carried out using the Python based toolkit *music21*. This toolkit allows users to carry out a huge spectrum of analysis on musical data along with facilitating the generation and editing of musical notation[9].

For each of the four songs in my dataset, I created a piano roll and histogram (Figure 1 and Figure 2 below shows examples of each).

---

[8] “jSymbolic 2.2,” SourceForge, accessed November 28, 2018, http://jmir.sourceforge.net/manuals/jSymbolic_manual/home.html.
[9] “What is Music21?,” MIT, accessed November 28, 2018, http://web.mit.edu/music21/doc/about/what.html.

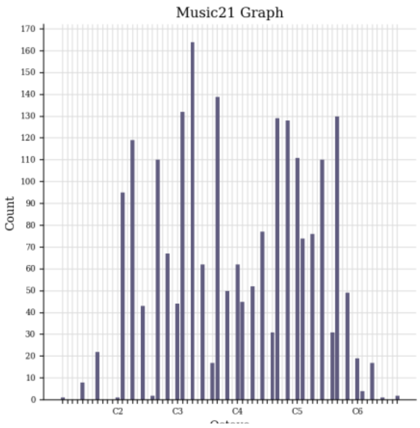

Figure 1 Histogram for "Hello"

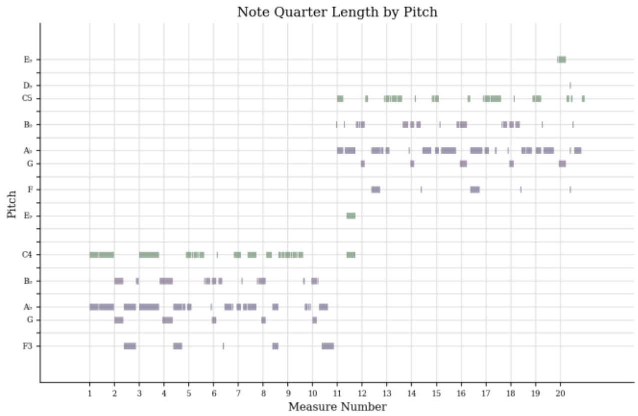

Figure 2 Piano Roll for "Hello"

The histogram shows how common various pitches are in the song and the piano roll shows the pitches present in each measure thus providing a very useful analytical visualisation.

Acoustic Data

In order to analyse acoustic data and look at music in the form of digital audio, due to copyright restrictions on Adele's material, I had to use royalty free music tracks from freemusicarchive.org. This limited my dataset as I could not provide comprehensive analysis of all three musical datatypes for the Adele material however I did manage to investigate different analytical techniques and produce data and visualisations from acoustic data.

Waveform and Time-Frequency Analysis

The first task I carried out was collecting technical and non-technical metadata for the files that I had downloaded. Using Sonic Visualiser (a software for viewing and analysing digital audio files), I established and recorded technical metadata such as the number of channels, sample rate and bits per second along with non-technical metadata like track title, genre and copyright information.

Using the same software, I created spectrograms (Figure 3) in order to carry out time-frequency analysis. The waveform visualisation (Figure 4) shows the frequency and amplitude of pitches over time whereas the spectrogram shows the frequencies that make up the sound over time[10] in more detail. In comparing these two visualisations, I identified that an advantage of time-frequency analysis over waveform is that general trends in certain frequencies are much easier to identify as waveforms show peaks in strength but it is hard to ascertain which frequencies are most prominent.

[10] "Spectrogram," Chrome Music Lab, accessed November 28, 2018, https://musiclab.chromeexperiments.com/Spectrogram/.

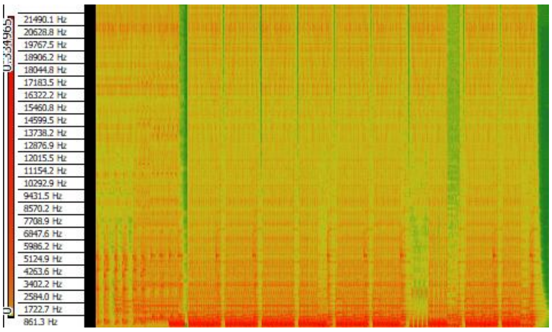

Figure 3 Spectrogram for "Hallon"

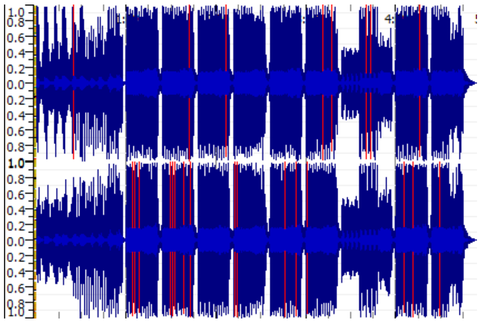

Figure 4 Waveform for "Hallon"

Extracting Higher Level Metadata and Meaning from Audio

Notated data typically contains explicit information about pitch, rhythm and time however recorded music tends to only store samples thus extracting higher level meaning can be difficult. With the help of specialist software, it is possible to extract this high-level information that can aid cataloguing, description and searching.

In Week 9's task, I used Sonic Visualiser to extract and produce Spectrograms (which show the strength of frequencies against time), Mel Frequency Cepstral Coefficients (which can be used to discriminate different instruments in a song) and Chromagrams (which shows the frequency of individual notes being played at a particular time) for three tracks. This extracted data can be used to facilitate analysis, comparison, transcription and very accurate searching based on criteria.

Using the MFCC data and Python, I then computed histograms for the three tracks (Figure 5). The histograms are created by scanning through each of the 20 MFCC bands and counting how often different energy levels occur. Using these, I could identify notable differences between songs and genres through looking at skewness, centres, flatness etc. In Figure 5, the histograms show relatively normal distribution with some having a slight, right skew[11].

In my dataset, I made the raw MFCC and Chromagram data files available and also displayed the three forms of analysis in their graphical forms along with the histograms.

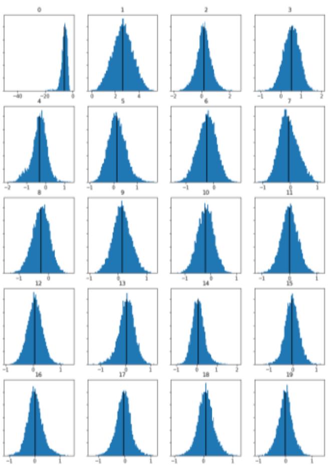

Figure 5 Histogram for "Farewell"

[11] "Typical Histogram Shapes and What They Mean," ASQ, accessed November 28, 2018, http://asq.org/learn-about-quality/data-collection-analysis-tools/overview/histogram2.html.

Similarity and Transcription

The final part of this project involved looking at applications of the data previously created in similarity-based searching and transcription. For the similarity-based searching analysis, a Python notebook was used that imported the Chroma features and through several different commands exported Feature Plots for a selection of songs, Mean Chroma Features graphs and finally Similarity matrices. Using Euclidian distance, the Python program calculates the distance between 10 tracks' vectors that are loaded in and exports them as a 10x10 matrix (Figure 6).

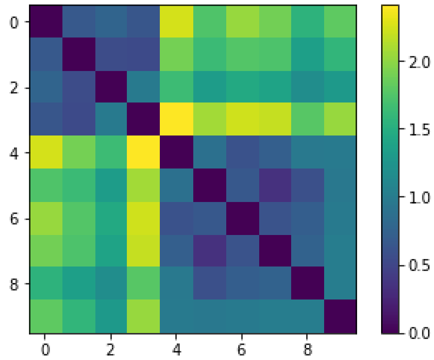

Figure 6 Similarity Matrix

As I could not use the studio recordings of Adele to carry out acoustic analysis, I did however export the MuseScore notated files as wave files. I then used these files to create Chroma Features plots which show the frequency of notes over time and Mean Chroma Features graphs which show which notes are present at a particular time in each song. I done this in order to create some acoustic analysis relating to my dataset theme.

The final part of Week 10's task involved investigating the quality of MuseScore's automatic transcription system. I firstly downloaded a completed MuseScore composition and exported it as a wave file. Using Sonic Visualiser, I then generated a piano roll and exported it as a MIDI file. On loading the MIDI file into MuseScore, it could be seen that the whilst the software attempted to create an accurate score, it did not have much success. Figure 7 shows that whilst the system produced a notated score that sounded close to the original, it visually was not user friendly and was near unreadable.

Figure 7 Sheet Music Comparison

Potential for Improvement

Whilst I feel that my project dataset contains a vast range of different analysis and content and is displayed in a very accessible and logical way, there are some improvements I feel I could have made:

- Transcribe more notated sheet music in order to have more pieces to analyse and compare in order to discover more accurate and interesting trends.
- Encode the MEI files with more comprehensive and useful metadata with @authURI and @xml:id attributes in order to make them more accessible.
- Choose a theme that has royalty free acoustic data in order to be able to complete a whole dataset containing data in the three forms.

Conclusion

In this project, I have compiled a vast amount of musical data in descriptive, notated and acoustic form. Several different methods of analysis have been used to produce interesting insights and visualisations with these techniques also being evaluated in this report. Whilst a relatively small selection of songs were encoded, I feel this dataset reflects a plethora of data creation and analysis.

BIBLIOGRAPY

"Advantage of MIDI." steinberg.help. https://steinberg.help/cubase_pro_artist/v9/en/cubase_nuendo/topics/midi_editor_score_editor/working_with_music_xml/score_editor_midi_advantages_of_c.html (accessed November 28, 2018)

"Get Audio Track Features for a Track." Spotify for Developers. https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/ (accessed November 28, 2018).

"jSymbolic 2.2." SourceForge. http://jmir.sourceforge.net/manuals/jSymbolic_manual/home.html (accessed November 28, 2018).

"MIDI." Wikipedia. https://en.wikipedia.org/wiki/MIDI (accessed November 28, 2018).

"Music Encoding Initiative." MEI. https://music-encoding.org/ (accessed November 28, 2018).

"MusicXML: Version 3.1." W3C. https://www.w3.org/2017/12/musicxml31/ (accessed November 28, 2018).

"Spectrogram." Chrome Music Lab. https://musiclab.chromeexperiments.com/Spectrogram/ (accessed November 28, 2018).

"The MEI Header." MEI. https://music-encoding.org/guidelines/v3/content/header.html (accessed November 28, 2018).

"Typical Histogram Shapes and What They Mean." ASQ. http://asq.org/learn-about-quality/data-collection-analysis-tools/overview/histogram2.html (accessed November 28, 2018).

"What is Music21?." MIT. http://web.mit.edu/music21/doc/about/what.html (accessed November 28, 2018).

Good, Michael. "MusicXML for Notation and Analysis." Songs and Schemas. http://michaelgood.info/publications/music/musicxml-for-notation-and-analysis/ (accessed November 28, 2018).

Logan, Beth. "Mel Frequency Cepstral Coefficients for Music Modeling." In *ISMIR*, vol. 270, pp. 1-11. 2000.

Schuller, Björn W. *Intelligent audio analysis*. New York: Springer, 2013.